

Use of Job Tickets to Secure Resource Access

Technical Field

The technical field is integration and control of services in a networked environment.

Background

Services may be provided by one or more operating units in a computer-based network. Users of the network may generate specific tasks and send the tasks into the network to be assigned to one of the operating units. For example, a user at a computer terminal may generate a printing order using a printer driver installed on the terminal. The printer driver is used to control the printing request. In another example, a user at a computer terminal may generate a printing order and send the printing order into a computer network so that the printing order is completed by a printing service. The printing order may be related to a company brochure. The printing order may contain unique requirements such as paper type, font size, layout, graphics, color, and other requirements. The user may specify that a specific printing service, such as Kinkos, prepare the company brochure. Alternatively, the computer network may include programs that suggest printing services to the user.

To control the printing job, the user's computer terminal may generate a job ticket. The job ticket includes the requirements, such as the requirements listed above, and an identification of the specific job that allows the job status to be tracked through the computer network.

Use of the job ticket allows printing and similar services to be allocated to those resources (i.e., the operating units) that are best suited to completing the services. Unfortunately, current computer systems do not allow access to the wide variety of services existing in networked computer systems, such as the Internet. In addition, current systems require users to have some knowledge of the existing resources, and may require users to include applicable programming to communicate with the services. Furthermore, current systems do not allow a job request to be split among several processors. As a result, completion of the job request may take longer than necessary, and may not be completed in the most efficient, lowest-cost manner. Finally, current systems do not provide adequate security for job tickets.

Summary

To overcome these and other problems related to use of a job ticket, a method and an apparatus allow a client to manage job attributes and processes using an electronic service center. The service center includes a job ticket service that allows access and modification of a job ticket by multiple users on a network. The method and apparatus use a network-accessible job ticket to relate to a specific job or content. The job ticket may be an object, such as an XML object, comprising routines and data. The content may be stored on the network and may be accessed by multiple job tickets. Storage and management of the job ticket are transparent to the user. The job ticket is stored in a common location in the network. The job ticket remains in the same location in the network, and users access only that portion of the job ticket required to complete a designated process. Security measures may be added to limit access to those users designated as being allowed to access the job ticket and the job file. The job ticket may include a service ID that relates the job ticket back to the originating job ticket service. In this way, a user who acquires all or part of the job ticket can refer back to the originating job ticket service (and the original, or as-modified, job ticket) to verify any changes and to ensure that the job ticket being accessed is up-to-date. The job ticket also includes a job ID to refer the job ticket to a specific job.

The service center is coupled through a communications network to a front end service. The front end service allows a user to generate a service or job request. The communications network may be the Internet, or a local area network, for example.

The service center includes a service bus, to which are coupled a job store, the job ticket service, and a work flow controller. Also coupled to the service center are one or more processors that may be controlled to complete processes and tasks defined in the job tickets.

The job ticket service may generate and store the job tickets. Job content (e.g., a PDF file) is stored in the job store. With this structure, the user does not have to manage storage of the job content or to know which job store holds the job content. The job ticket service controls access to the job tickets, and, through the use of the job tickets, also controls access to job content in the job store, or elsewhere in the network. The job ticket service may create a reference to the job ticket, and may use the reference to control access to the job ticket.

A job ticket service center may include several features to provide security and to control access to the job ticket. In an embodiment, the job ticket service may employ branch locking. In

an alternate embodiment, servers may be used to authorize and authenticate a processor. An authentication server receives authentication information from a processor and an authorization server uses the information to check authorization functionality. The authorization or access rights of the processor may be carried as a part of the job ticket. Using the above-described features, the service center may provide trusted authentication information about the processor to the authorization server, and the authorization server then performs its authority check functions using this information.

In an embodiment, the job ticket may be signed with an industry standard public message digest (MD) signature, and may be protected by a public key encryption system. Hence, any entity that has the public key may validate the job ticket without having to communicate with the authentication server. These features reduce communication between distributed server applications. The features also allow the job ticket to be passed from one processor to another processor, maintaining security, without communicating with the service center.

In an embodiment, when a processor modifies a branch, the processor accesses the job ticket service to update the job ticket. The modified job ticket may then be signed before being passed to the next processor.

In yet another alternative embodiment, the job ticket holds authentication/access data, allowing controlled access to the job ticket. Resources may be protected by passwords and other mechanisms. Access to the job ticket may be similarly protected. Furthermore, processors with access authorization may have such access authorization invoked by listing the processors in the job ticket. The listing may be effectuated by recording a network address for the processors, for example. The network address may be incorporated in bid information recorded in the job ticket.

Description of the Drawings

The detailed description will refer to the following figures in which like numerals refer to like items, and in which:

Figure 1 is a block diagram showing a prior art use of a job ticket;

Figure 2 is a tree diagram showing the processes in an example job ticket;

Figure 3 is a block diagram of a digital image work flow network;

Figure 4 is a block diagram of a service center used with the network of Figure 3;

Figures 5A-5D illustrate an exemplary job ticket;

1 Figure 6 is a diagram of functions controlled by a job ticket service;
2 Figure 7 is a diagram showing access functions controlled by the job ticket service;
3 Figure 8 is a block diagram illustrating additional control features of the job ticket service;
4 Figure 9 is a flow chart illustrating one of the processes controlled by the job ticket service;
5 and
6 Figures 10-15 are flow charts showing sub-processes in the overall process illustrated in
7 Figure 9.

8 Detailed Description

9 Figure 1 is a block diagram showing a prior art application of a job ticket service. Job
10 tickets are often associated with a printing standard, the job definition format (JDF). The JDF is
11 described in detail in JDF Specification Draft Spiral 4.0, available at www.hp_opensource.com,
12 which is hereby incorporated by reference. In Figure 1, a user 1 generates a job request and sends
13 the job request through a portal 4 to a processor 5. The job request may include a job ticket data
14 file 2 and a content file 3. The user 1 may be a computer terminal in a networked computer system
15 and the processor 5 may be a networked printer. The job request may involve printing a
16 document. The document may be represented by the content 3, which is a digital representation
17 of text and images to be printed. The intended format of the printed document may be described
18 in the job ticket file 2, which is simply a digital file that specifies how the printer is to print the
19 document. For example, the job ticket file 2 may require that the document be printed on back-to-
20 back pages.

21 In a specific application, the functions of the job ticket file 2 may be carried out by a printer
22 driver. The printer driver encodes control data related to printing the document, and sends the
23 control data and the content 3 to the printer (i.e., the processor 5). The printer accesses the control
24 data and the content 3 to print the document.

25 While the application shown in Figure 1 works well to print a document, the application has
26 many drawbacks. In particular, if multiple processors are involved in producing the document,
27 each such processor will require access to the job ticket file 2. This access brings problems related
28 to security, modification control and workflow control. For example, each processor requiring
29 access to the job ticket file 2 may have to wait on processing until a prior processor has completed

1 use of the job ticket file 2. Thus, the prior art application may result in unwanted delays in
2 completing the job request.

3 Prior art applications of job ticket services also suffer because the user may not know
4 anything about the processors, including capabilities and availabilities of the processors, or even if
5 the processors exist. Thus, the user may not know which portal to use to connect to a specific
6 processor.

7 These and other problems are solved by a method and an apparatus that controls access
8 to a job ticket and associated content through use of a job ticket service. The job ticket service
9 includes mechanisms that arbitrate access to the job ticket among multiple users of the job ticket,
10 limit access to the job ticket by incorporating security features, and ensure modifications made by
11 one processor or user are reflected in the job ticket and the content. In effect, the apparatus
12 includes a generic database that couples input data from clients as job requests with output services
13 such as processors that perform tasks or processes to complete the job requests. The database
14 may have the features of a generic XML database in that it is extensible, and in that the clients need
15 not have any knowledge of the individual processes to be performed, or the internal programming
16 requirements of the processors. Thus, the clients may submit job requests to a service center that
17 will ensure that an appropriate processor or processors are assigned to complete the job request.

18 Before describing the apparatus and method in detail, a review of a job ticket is provided.
19 Figure 2 is a node-tree diagram (or simply a node tree) 10 that illustrates processes defined in a
20 job ticket for printing a brochure. The brochure may be printed on a commercial press, and may
21 use digital content to generate plates for printing the brochure. Within the node tree 10, the nodes
22 specify a product, process, or group of processes. Each node may modify, consume or create
23 resources. Each node may contain further nested nodes, or sub-nodes. The arrangement of nodes
24 and sub-nodes may be likened to a tree, and each node and sub-node may be referred to as a
25 branch. A brochure node 11 defines the features and parameters of the brochure. A cover node
26 12 defines the parameters for producing the brochure cover. Inside pages node 13 includes the
27 parameters to produce the inside pages. The inside pages node 13 is shown with several sub-
28 nodes, including a sub-node 14 for digital plate making. The digital plate making sub-node 14 itself
29 includes two additional sub-nodes, a ripping sub-node 16 and a plate making sub-node 18.

Each of the nodes and sub-nodes shown in Figure 2 has associated with it input resources and at least one output resource. A resource may be described by parameters or logical entities. The resource may be a physical entity such as a component, a handling resource, or a consumable. A component resource may be the output of a node or sub-node, such as a printed sheets. A handling resource is used during a process, but is not consumed by the process. A consumable resource may be partly or wholly consumed by the process. Examples of consumable resources include inks, plates, and glue. Other resources may be a digital file or representation of a physical object. For example, the ripping sub-node 16 may include as input resources a run list, media, RIP parameters, and layout. The run list resource describes the pages, including the files in which the pages occur, and which pages are to be used. The media resource describes the media that will be used to make plates, and is needed to describe the dimensions of the media. The RIP parameters resource describes all device-specific parameters of the ripping process. The layout resource describes placement of source pages onto the plates, and eventually onto press sheets. As an output resource, the ripping sub-node 16 may provide ripped flats. Other resources include parameter resources, which define the details of processes, as well as other non-physical computer files used by a process.

The node tree 10 shown in Figure 2 is intended to apply to printing a document. However, node-tree diagrams may be used to represent job tickets for other services besides printing. For example, a job ticket may be used for data processing, image processing, creating and maintaining a database, electronic publishing, e-mail, and various e-commerce services. Moreover, the job ticket may be used to allow different e-commerce services to interact with each other.

Figure 3 is a block diagram of a digital imaging work flow (DIW) network 20 that incorporates a service center and a job ticket service to control tasks submitted by clients. The service center may operate as a single portal through which the clients connect to one or more e-services including e-mail, e-commerce and online shopping, e-printing, and data services, including database searching and database construction, population and maintenance. In an alternative embodiment, the service center may comprise multiple portals, with each of the multiple portals being dedicated to a specific e-service, or being provided to increase bandwidth. Using a single portal, such as the service center, allows the clients to select from a wide variety of e-services, such as those noted above, without requiring the clients to have any prior knowledge of the e-services.

1 The service center may include components that receive information in the form of job
2 requests, and using the information, create a job ticket that specifies tasks and resources. The job
3 ticket may be stored in a job ticket service, and notices may be posted to indicate when a job ticket
4 is available. Processors coupled to the service center may bid on completion of the job ticket, and
5 the service center may include a bidding service that evaluates bids. The service center may select
6 one or more processors to assign to the job ticket based on client-supplied criteria, or based on
7 a set of standard criteria, including industry standard criteria. The service center may provide
8 mechanisms to control access to the job ticket, or to portions (branches) of the job ticket. The
9 mechanisms include branch locking, and authorization and authentication servers that use public key
10 encryption, or similar processes.

11 The service center may include hardware components such as servers, computers, central
12 processing units, communications interfaces, and memory devices to provide the processing
13 capability and data storage required to carry out the above-described functions.

14 The DIW network 20 includes a front end service 30 that allows a client 31 to generate
15 and submit a service or job request 32 (see Figure 7). In an embodiment, the front end service 30
16 may be an Internet web browser. Alternatively, the front end service 30 may be a web application
17 or a port monitor. The job request 32 may contain detailed information about how the job is to be
18 executed, and may be formatted according to the job definition format standard. Alternatively, the
19 job request 32 may include only basic information, which will be used by another component to
20 finalize the job definition, or work flow. Finally, the job request 32 may include the content, or job,
21 that is to be processed. The content could be one or more digital files, text files, and other files.
22 The front end service 30 is coupled to a communications network 35, which may be the Internet
23 or a local area network, for example. Coupled to the communications network 35 is a service
24 center 40 that links one or more processors 80, to the communications network 35. Each of the
25 processors 80_i may include a cache 81_i that may be used to store information related to a job
26 request 32, including information related to job tickets. In an embodiment, the service center 40
27 may be an Internet web site that includes data storage and control functions. In another
28 embodiment, the service center 40 is a node in a local area network.

29 The service center 40 allows a broad spectrum of communications between entities coupled
30 to the service center 40. In particular, the service center 40 allows different e-services to interact

programmatically with one another using specific protocols and generic protocols (e.g., TCP/IP). This programmatic interaction allows different services and processes that are coupled to the network to exchange data and files, and to modify the data and files. The programmatic interaction may be completed by use of a remote procedure call (RPC) between entities coupled to the service center 40. Other methods for providing the programmatic interaction include CORBA, UDDI, and e-speak.

Figure 4 is a diagram of the service center 40. The service center 40 includes a service bus 41 in communication with the communications network 35 and the processors 80_i of Figure 3. Coupled to the service bus 41 is a job store 50, a job ticket service 60, a work flow controller 70, an optional bidding service 90, an authorization server 92 and an authentication server 94. The job store 50 may store one or more job content files 51_i. The job ticket service 60 may control one or more job tickets 61_i. The work flow controller 70 may use one or more agents 71_i to control processes on the service bus 41.

The job store 50, job ticket service 60 and work flow controller 70 function to accept information from the clients 31, and to use the information to control the actions of the processors 80_i. The processors 80_i performs specific tasks or processes as determined by the service center 40.

The job store 50 may be a node on the service bus 41, and may include programming to allow the job store 50 to carry out its functions. The job store 50 may be used to store the content 51, which may be in the form of one or more large files. In the context of printing a document using a service or process coupled to the service bus 41, the job store 50 may store the document content in one or more PDF files, for example. The content 51 may include graphics and text. The content 51 for a specific document may include several files. For example, a brochure may have a separate file for the cover and another file for the inside pages. Text for the inside pages may be in one file and images in yet another file. The content 51 may also include links to other resources or entities on the service bus 41. The job store 50 provides for mass storage of the content 51, so that a user (client 31 or processor 80) does not have to provide the mass storage required for the job content 51. By using the mass storage capabilities of the job store 50, the content 51 may be made to persist in the network 20, and may be made accessible to users at any time. The job store 50 also manages and controls the content so that the user (client 31 or processor 80) does not have

to manage the content 51. Management functions include maintaining configuration or version control of the content 51, controlling access to the content 51, and maintaining the content 51 in storage.

The job ticket service 60 holds job tickets 61. The job ticket service 60 controls access to and may manage configuration of the job tickets 61. For example, the job ticket service 60 may allow users (clients 31 and processors 80) to access a portion or branch of a job ticket 61 rather than passing the job ticket 61 among multiple users. Access to the job ticket portion may be effectuated by use of an application programming interface, a scriptable interface, or a similar feature. As noted above, the job ticket 61 does not include the content 51 (e.g., the graphical and text files of a document), but the job ticket 61 relates to content 51 (e.g., a PDF file) stored in the job store 50. The user does not have to manage storage of the job content or to know which job store 50 holds the job content. The job ticket service 60 instead passes a reference in the job ticket 61. This allows multiple clients 31 and processors 80, to access the content 51. Furthermore, the content 51 may relate to more than one job ticket 61. The job ticket service 60, and its interrelationships with other entities coupled to the service bus 41, will be described later in detail.

Some job tickets 61 may be accessed by multiple processors 80, in either serial, overlapping, or simultaneous fashion. The multiple access processing could result in problems with use of the job ticket 61. For example, a first processor may acquire the job ticket 61 (or a portion or branch thereof), and perform a process specified in the work flow, which may modify the branch. Such modification may be to indicate a branch as complete, use up input resources, or create new output resources, for example. A second processor could attempt to acquire the branch, but might not “know” that the first processor had modified the branch. Alternatively, if two processors compete for the same branch, a deadlock situation might occur.

One solution to the above problems may be to lock the job ticket 61 whenever a processor 80 acquires the job ticket 61. Unfortunately, locking the job ticket 61 may prevent concurrent or parallel processing and may slow down completion of the job request 32.

The job ticket service 60 shown in Figure 4 overcomes these and other problems by having the capability to lock the job ticket 61 at the branch level. The branch locking may be accomplished by one of several methods. The work flow controller 70 may assign one or more

specific processors 80_i to perform the tasks identified with the branch to be locked. Where only one processor 80 is authorized access to the branch, branch locking may not be required. Where more than one processor 80 is authorized access to the same branch, the job ticket service 60 may lock the branch when one of the authorized processors 80_i actually acquire the branch.

If the work flow controller 70 has not assigned processors 80_i to branches (i.e., any processor 80 may access a branch at any time), the job ticket service 60 may lock the branch when a processor 80 acquires the branch.

The job ticket service 60 may lock the branches by setting a lock/unlock flag for each branch. Processors 80_i accessing the job ticket 61 may then review the lock/unlock flag status to determine if the branch may be accessed. In some circumstances, the job ticket service 60 may allow access only to those branches that are unlocked. A processor 80 that has completed a task defined by the branch may need to have the branch unlocked in order to modify the branch.

The work flow controller 70 may be used to create the job tickets 61 that are stored in the job ticket service 60. The work flow controller 70 may review the job requests 32 submitted by the clients 31, and may then use a job ticket template to prepare the job ticket 61. The work flow controller 70 may then send the job ticket 61 to the job ticket service 60 for storage and processing.

The work flow controller 70 also controls completion of tasks among the processors 80_i. In an embodiment, the work flow controller 70 determines which of the processors 80_i have the necessary and available resources to begin the processes listed in a specific job ticket 61. The work flow controller 70 then designates the appropriate processors 80_i to complete the tasks referenced by the job ticket 61. For example, if a job ticket 61₁ requires color printing, the work flow controller 70 may determine that only processor 80₃ is a color printer with the capacity to begin the job specified in the job ticket 61₁. This embodiment in which the work flow controller 70 determines which processors 80_i to assign to a specific job ticket 61 may be especially appropriate when the network 35 is a local area network and all processors 80_i are directly coupled to the local area network 35.

Alternatively, the work flow controller 70 may receive bid information from Internet-connected processors 80_i and may use the bid information to select the processors 80_i to complete the job request 32.

1 The work flow controller 70 may also be used to designate the various nodes, input and
2 output resources, and other features of the node tree used to complete the job request. That is, the
3 work flow controller 70 may be used to create a construct, or work flow, such as the node tree
4 10 shown in Figure 2. To accomplish these tasks, the work flow controller 70 may include one or
5 more agents 71 that write a job definition file, based on control data contained in the job request
6 32. Alternatively, a separate management information system (not shown) may be used to create
7 the nodes, and to control flow of tasks to the processors 80_i and other entities. In yet another
8 embodiment, the job definitions may be written by the client 31 that originated the job request 32.

9 Referring again to the node tree 10 of Figure 2, many output resources of the individual
10 nodes serve as input resources for other nodes. These other nodes may not be able to begin
11 executing until all input resources are complete and available, which means that the nodes may need
12 to execute in a well-defined sequence. For example, a process for making plates will produce
13 press plates as an output resource that is required by a printing process. In the hierarchical
14 organization of the node tree 10, nodes that occur higher in the node tree 10 represent higher-level,
15 more abstract operations, while lower order nodes represent more detailed, specific processes.
16 Moreover, nodes near the top of the node tree 10 may represent only intent regarding the
17 components or assemblies that comprise the product, and lower level nodes provided the detailed
18 instructions to a processor 80 to perform a specific process.

19 Because two node trees may not be similar, the work flow controller 70 may determine
20 processes to be completed, the order in which the processes are completed, and the processors
21 80_i that are to complete the processes. The work flow controller 70 may use the agents 71 to
22 determine an actual work flow, considering factors such as control abilities of the processors 80_i,
23 that complete the processes, transport distances between processors, load capabilities of the
24 processors 80_i, and time constraints in the job request, for example. The agents 71 may define the
25 overall process using serial processing, which involves subsequent production and consumption of
26 resources by the processors 80_i, overlapping processing, which involves simultaneous consumption
27 and production of resources by more than one processor 80, parallel processing, which involves
28 sharing resources among processors 80, and iterative processing, which involves a back and forth
29 processing scheme to develop resources.

8 As an alternative to polling, processors 80_i coupled to the service center 80 may monitor
9 the job ticket service 60. The job ticket service 60 may periodically post, in a bulletin board
10 fashion, for example, notices for job tickets that are available for processing. The processors 80
11 may then submit a bid for the tasks and processes defined in the job ticket notice. The work flow
12 controller 70, or the separate, optional bidding service 90, may review the bids, and determine
13 which single processor 80 or combination of processors 80_i would be best suited to complete the
14 tasks and processes defined in the job ticket notice.

The service center 40 may include several features to provide security and to control access to the job ticket 61. As discussed above, the job ticket service 60 may include a provision for branch locking. In addition, servers may be used to authorize and authenticate a processor 80 and maintain the authorization and authentication during completion of a job request 32 (see Figure 7). The authentication server 92 receives authentication information from a processor 80 and the authorization server 94 uses the information to check authorization functionality. The authorization or access rights of the processor 80 may be carried as a part of the job ticket 61. The servers 92 and 94 may be hardware devices, but need not exist in the same hardware platform, and the servers 92 and 94 need not be tightly coupled. Alternatively, the functions of the servers 92 and 94 may be performed in programming stored in one of the components of the service center 40, such as the work flow controller 70, for example. Using the above-described features, the service center 40 may provide trusted authentication information about the processor 80 to the authorization server 94, and the authorization server 94 then performs its authority check functions.

28 The job ticket 61 may be signed with an industry standard public key encryption message
29 digest (MD) signature, and may be protected by a public key encryption system. Hence, any user
30 that has the public key may validate the job ticket 61 without having to communicate with the

1 authentication server 92. These features reduce communication between distributed server
2 applications. The features also allow the job ticket 61 to be passed from one processor 80 to
3 another processor 80, maintaining security, without communicating with the service center 40.

4 In an alternative embodiment, the job ticket 61 holds authentication/access data, allowing
5 controlled access within the service center 40 infrastructure. Resources may be protected by
6 passwords and other mechanisms. Access to the job ticket 61 may be similarly protected.
7 Furthermore, processors 80, with access authorization may have such access authorization invoked
8 by listing the processors in the job ticket. The listing may be effectuated by recording a network
9 address for the processors 80, for example. The network address may be incorporated in the bid
10 information recorded in the job ticket 61.

11 Although the above description refers to development by the work flow controller 70, other
12 components in the network 20 may be used to develop an overall work flow to complete the job
13 request 32. For example, the job ticket service 60 may be used to develop the overall work flow.

14 As discussed above, the bidding service 90 may be used to receive bid information from
15 processors 80, coupled to the service center 40. The processors 80, submit bids in response to
16 posting of job ticket notices at the service center 40. In an embodiment, the job ticket notice is a
17 separate object stored in the service center 40. In another embodiment, the job ticket 61 itself
18 serves the notice function. The work flow controller 70 may post the job ticket notices after receipt
19 of the job request 32. Whether the bidding service 90 or the work flow controller 70 receives the
20 bids, the bid evaluation and selection process may be the same.

21 The job ticket notice posted by the work flow controller 70 may include specific tasks or
22 processes (branches) that must be completed to complete the job request 32. A simple job request
23 32 may have only one branch. More complex job requests 32, such as the job request illustrated
24 in Figure 2 (i.e., print a brochure) may have many branches. Furthermore, some branches may be
25 so interrelated that they can only be completed in a specific sequence, while other branches can be
26 completed in a parallel or an overlapping fashion. This interrelationship may often be the result of
27 one branch producing an output resource that is an input resource for one or more other branches.
28 The job ticket notice may include descriptions of specific branches and their interrelationships in
29 sufficient detail to allow the processors 80, to bid for completion of the branches. The job ticket
30 notice may persist in the service center 40 for a specified time to allow the processors 80, to send

1 bids. The time may be a set value (e.g., one hour) or may be based on a completion deadline
2 specified in the job request 32.

3 The bidding service 90 may select bids 91 from the processors 80 based on set criteria.
4 For example, the job request 32 may specify minimum performance requirements (e.g., a maximum
5 cost and a completion deadline). The bidding service 90 may reject any bids that fail to satisfy the
6 minimum performance requirements. Where the work flow controller 70 has established multiple
7 branches, each such branch may include minimum performance requirements. The branch-specific
8 performance requirements may be established by the work flow controller 70 based on overall
9 performance requirements for the job ticket 61. A processor 80 that bids on a particular branch
10 may be rejected by the bidding service 90 if the processor 80 fails to meet the minimum
11 performance requirements.

12 If the client 31 does not specify any minimum performance requirements, the bidding
13 service 90 may apply a standard set of criteria (e.g., an industry standard). In addition, the bid
14 must satisfy any requirements for producing output resources. In this way, bids that are made in
15 error, or that would otherwise likely be rejected, can be screened out. For example, a bid for
16 printing inside pages of the brochure may indicate a one year completion date. Such a bid may be
17 rejected, even in the absence of any specified performance requirements from the client 31.

18 In addition to submitting performance requirements, the client 31 may specify an evaluation
19 algorithm for evaluating bids. For example, the client 31 may specify that cost is to be weighted
20 twice as much as any other performance requirement.

21 In the absence of a client-specified evaluation algorithm, the bidding service 90 may apply
22 a standard evaluation algorithm in order to rank bids for each branch in the work flow. The
23 evaluation algorithm may apply weighting criteria, or may apply a default rule. For example, bids
24 may be ranked based on a maximum score, where points are awarded for cost estimates below
25 a maximum and for completion times below a maximum. Once the evaluation algorithm has been
26 applied, the bidding service 90 ranks the bids for each branch. If only one processor 80 survives
27 the process, that processor 80 may be automatically selected and assigned to the branch. If
28 multiple processors 80_i survive, the bidding service 90 may provide a list of such processors 80_i
29 to the work flow controller 70, which will then select the processors 80 to be assigned to the

branches. Alternatively, the list may be provided to the client 31, and the client 31 may select the processor(s) 80_i to complete the tasks defined in the work flow.

The work flow controller 70 may associate winning bids with corresponding branches, and may store the bid information with the job ticket 61. The stored bid information may include identification information that allows the authorization server 94 and the authentication server 92 to permit access to job ticket branches or to the entire job ticket 61. Because the bid information is stored with the job ticket 61, a processor 80 may access those branches for which the processor 80 is authorized access without having to communicate directly with the job ticket service 61. This feature allows the job ticket 60 to be passed from one processor 80 to another processor 80, which improves processing time and efficiency.

In an embodiment, the work flow controller 70 accesses control data of the job ticket 61 to determine which processor(s) 80_i should be assigned to the specific task identified in the job ticket. The work flow controller 70 may also identify which of the processors 80_i would be able to meet the criteria specified in the control data, and may provide a list of such processors 80_i to the client through the front end service 30. The client 31 may then select a processor(s) 80_i from the list.

In an embodiment, the job ticket service may be embodied as a sequence of program instructions stored on a computer-readable storage device, such as a CD-ROM, for example. When loaded into, or coupled to a computer, the program instructions may be read and executed by a processor of the computer to provide the functions of the job ticket service.

Figure 5A illustrates an exemplary job ticket 61. The job ticket 61 may include two parts. A first part includes a framework 62 and an optional client extension 64. The framework 62 includes information, files and programming necessary to control tasks defined in the job ticket 61. The client extension 64 may include information related to a specific client (machine) and to a user of the machine. A second part includes a security module 67 that protects the job ticket 61 from unauthorized access.

The framework 62 may include a job identification (ID) 63, a service ID 65, a task section 68, and a control data section 69. The job ID 63 includes a reference to a specific job, or content 51 that is stored in the job store 50. The job ID 63 also includes a reference to a particular job store 50 that is used to store the content 51. An entity that acquires a reference to the job ticket

61 can use the job ID 63 to access the corresponding content 51. Thus, the network 20 shown in Figure 3 may include multiple job stores 50, and the job ID 63 may be used to correlate the job ticket 61 to a specific job store 50. The service ID 65 identifies a specific job ticket service 60 that stores the job ticket 61. For example, the network 20 may include multiple job ticket services 60 (not shown in Figure 3). The service ID 65 is used to correlate the job ticket 61 to the appropriate job ticket service 60.

The tasks section 68 (Figure 5B) may include branch definitions, and other information needed to control completion of the branches. The tasks section 68 may be structured so that each branch or node in a node tree is represented by one or more branches 66, in the tasks section. In this embodiment, each node in the node tree (e.g., the node tree 10 of Figure 2) can have associated with the node, the description 95, resources 96, lock/unlock flag 97, and security functions 99. In this way, the job ticket 61 reflects a hierarchical database structure.

The control data section 69 includes the specific instructions, parameters, and criteria for completing the task identified by the job ticket 61. Control data in the control data section 69 may also be associated with each node in a node tree.

The security module 67 controls access to a specific job ticket. The security module 67 may be implemented using standard encryption and access techniques, including public/private key infrastructures, for example. In an embodiment, the security module 67 may include a list of authorized processors 80, that may be given access to one or more branches of the job ticket 61, or to the entire job ticket 61.

The client extension 64 may contain "custom" information, such as user age, credit card number and zip code. Information provided in the client extension 64 may be protected by use of a public key signature, or similar feature. Hence, all client extension information will automatically be included in a Message Digest Protocol (MDP) and will affect the signature of the job ticket 61. With the above-describe job ticket architecture, many Internet-related security issues are addressed, including IP spoofing, time controlled sessions, job ticket alterations, varying authorization levels, and client-dependent persistent data storing.

The job ticket 61 shown in Figure 5A may be used to refer to a specific content 51 in the job store 50. Alternatively, multiple job tickets 61 may be used to refer a specific content 51, or one job ticket 61 may be used to refer to multiple contents 51. Thus, for example, one job ticket

3 Using the network 20 shown in Figure 3, and the corresponding job ticket shown in Figure
4 5A, a client 31 may request and have completed many different electronic services. For example,
5 the client 31 may use the network 20 as an e-mail application.

Figure 5B shows the tasks section 68 in detail. The tasks section 68 may include one or more branch descriptors 66 that include information related to processing for that branch. A description segment 95 may define the tasks to be completed for each branch. Alternatively, the description segment 95 may provide a link, or handle, to a file that contains the branch description. The resources segment 96 lists input and output resources associated with the tasks defined for the branch. The lock/unlock flag segment 97 allows a flag to be set to lock and unlock a branch. A bid information segment 98 includes bid information gathered, for example, by the bidding service 90. The bid information 98 may include detailed information such as the IP address of the processors authorized access to the branch, estimated performance information (e.g., estimated cost, delivery time), and other information. Alternatively, the bid information 98 may contain a link to another file containing the detailed bid information. The security segment 99 may indicate authorized security levels, and may be used as part of a public key/private key infrastructure.

Figure 5C illustrates an embodiment of the control data 69. The control data 69 includes a client address, which may be a machine address, such as an Internet protocol (IP) address. An expiration date/time segment may be used to terminate active status of the ticket 61. Once terminated, the ticket may be deleted from the job ticket service, and the corresponding content 51 may be de-referenced. That is, the content 51 may no longer be referenced by a specific job ticket 61. This feature may help eliminate stale data, and free up resources for other job requests 32 (see Figure 7). Finally, the control data 69 may include specific performance requirements, such as cost an delivery, warranty, required materials, price reductions based on quantity, and other requirements, for example.

The use of job tickets as XML objects allows clients to define databases, and to store data through the job ticket service 60 and the job store 50. The databases may be used to hold contact lists, addresses, and other personal data. The databases may also be used to store any other generic data. The databases could then be used in conjunction with a variety of e-services

1 completed. Staying with the example of the contacts database, the client 31 may desire to send
2 a facsimile transmission to all entries in the contact list that have a specific zip code. The job ticket
3 service 60 can search the contacts database by properties, looking for zip code. The job ticket
4 service 60 can also search the contacts database to determine if any entry does not have a facsimile
5 machine. For those entries that do not have a facsimile machine, the job ticket service 60 can
6 originate a message to send back to the client 31, informing the client 31 that the facsimile
7 transmission was undeliverable. Using this functionality, the client 31 need not know anything about
8 the intended recipients of the facsimile transmission.

9 Returning to the example of an e-mail service, at the client 31, an e-mail application may
10 be launched in order to send an e-mail message, using the Internet, to one or more contacts in the
11 contact database. However, the client 31 need not subscribe to any one Internet service provider.
12 Instead, the service center 40 determines which processor 80 best suits the client's needs for
13 sending the e-mail message. That is, the service center 40 may select a e-mail service provider (a
14 processor 80) to send the e-mail message to a chosen destination address. Furthermore, the
15 service center 40 may determine, based on information maintained in the contact database (i.e., the
16 content 51 in the job store 50), which delivery options are desired by a user at the destination
17 address. For example, the destination address user may desire that all e-mail messages be sent to
18 an e-mail box, or that an alert be provided whenever an e-mail message is sent. These delivery
19 features may be stored in the contact database. Alternatively, the delivery features may be stored
20 in a separate database (content file 51) in the job store 50, and the service center may retrieve
21 information from this separate database when determining how to deliver the e-mail message.
22 Specifically, the separate database may include a variety of users, along with the user's Internet
23 address. By comparing the Internet address provided with the out going e-mail to the Internet
24 addresses in the separate database, the service center 40 can determine desired delivery options
25 of the addressee. This process for determining delivery options is transparent to the client 31 that
26 originated the e-mail message. All that the client 31 need know is the contact information (e.g., the
27 Internet address).

28 The client 31 may use the job ticket service 60 to specify a number of performance features
29 related to the e-mail service. For example, the client 31 may want the service center to attempt a

1 specified number of delivery attempts, and if delivery does not occur, to send a return message to
2 the client 31 indicating non-delivery of the e-mail message.

3 As noted above, the job ticket 61, in conjunction with other components of the service
4 center 40, may also be used to create a persistent, generic object-based data structure, such as an
5 XML database. An example of the use of a job ticket 61 for this purpose is illustrated in Figure
6 5D. The job ticket 61 includes a contacts list 84, which may be in the form of an XML database,
7 or some other generic database. The contacts list 84 may include a structure with entries for
8 business 85 and personal 86 use. The business 85 and personal 86 contacts structures may include
9 entries of individuals 87, as shown. Each of the entries 87 may include specific properties, as
10 defined above. In addition, or alternatively, each of the entries 87 may include links to other
11 databases that provide additional information and properties about the individual.

12 While the use of the job ticket 61 as a XML database has been described with reference
13 to an e-mail and messaging service, the job ticket 61 is not so limited. Any data that is capable of
14 being stored in a database may be accesses and controlled using the job ticket 61.

15 The features described above, and shown in Figures 5A-5D, may be replicated in another
16 embodiment of a job ticket 61 in which all data related to a specific node or branch is located with
17 that node or branch. Using the example node-tree 10 shown in Figure 2, each node (branch) may
18 include detailed information and features such as resources, authorized processors 80, lock/unlock
19 flag, bid information, branch description, and other information.

20 Figure 6 is a diagram of functions of the job ticket service 60. The primary functions of the
21 job ticket service 60 are to store 73 the job tickets 61, and to provide access 75 to the job tickets
22 61, to users such as the client 31 and to the processors 80. To accomplish these storage and
23 access functions, the job ticket service 60 may create a job ticket reference 72 and a job resource
24 reference 74. The job ticket service 60 also controls job content access 76, updates 77 the job
25 tickets 61, as processes are completed and reported by the processors 80, completes the job
26 tickets 61, and reports 78 when all processes are completed for a specific job ticket 61, and
27 provides an approval process 79 to allow a client 31 to approve completion of the tasks designated
28 in the job ticket 61.

29 The job ticket reference 72 includes a specific reference to a corresponding job ticket 61.
30 The job ticket reference 72 may be used by the job ticket service 60 to allow one or processors

80_i and clients 31_i to access the job ticket 61. That is, instead of passing the job ticket 61 to a processor 80, the job ticket service 60 passes the job ticket reference 72. With the job ticket reference 72, the processor 80 may access all or a part of a job ticket 61 so that the processor 80 may complete one or more processes. Unlike conventional job ticket services, the job ticket service 60 retains the job ticket in storage 73, and only permits users (clients 31_i and processors 80_i) to access the job ticket 61. This feature allows multiple processors 80 to simultaneously complete processes for the specific job request 32 related to the job ticket 61.

The job ticket service 60 may also create a resources reference 74, and may provide the resources reference 74 to the processors 80 and the clients 31 in a manner similar to that of the job ticket reference 72. As noted above with the description accompanying Figure 2, the resources may include physical devices and materials, and may include digital files. Use of the resources reference 74 may simplify data included in the job ticket 61.

Alternatively, information contained in the resources reference 74 may be included in the job ticket 61, or may be included in other files accessed by the clients 31_i and the processors 80_i.

Figure 7 is a diagram showing operation of selected functions of the job ticket service 60. As shown in Figure 7, the job ticket service 60 includes a job ticket 61_i, which may be a programming object such as that represented in Figure 2, and described above. The job ticket 61_i is shown supplied to the job ticket service 60 by the client 31_i. The client 31_i may be a networked computer or similar device that is capable of transmitting the digital information representing the job ticket 61_i to the job ticket service 60. To ensure the job ticket 61_i arrives at the job ticket service 60, the job ticket 61_i may contain a reference to the job ticket service 60, such as the service ID 65 illustrated in Figure 5B. The service ID 65 may include a network address of the job ticket service 60. For example, the service ID 65 may include a universal resource locator (URL) if the job ticket service 60 is an Internet web site.

Also shown in Figure 7 are client 31₂ and processors 80₁ - 80_N. The processors 80₁ - 80_N may include networked resources such as networked printers, electronic-commerce entities, such as Internet web sites, and "brick and mortar" entities, such local print shops that are coupled to the job ticket service 60 using the service bus 41.

The client 31 generates a job request 32 (content 51 and job ticket data). Using the front end service 30 (not shown in Figure 7) and the service bus 41, the client 31_i sends the job ticket

1 data to the job ticket service 60 and the content 51 (not shown in Figure 7) to the job store 50.
2 The job ticket service 60 may pass the job ticket data to the work flow controller 70, which will
3 create a job ticket 61. The content 51₁ and the job ticket 61₁ are related by the job ID 63. The
4 job ID 63 also includes an identification of the job store 50, and a location within the job store 50
5 in which the content 51₁ is stored. In an alternate embodiment, the content 51₁ may be stored at
6 the client 31₁, and may then be accessed by other users through the service bus 41 and the front
7 end service 30.

8 The job ticket 61₁ specifies processes that must be completed to finish the job request 32.
9 As noted above, Figure 2 illustrates processes required to print a brochure, including the inside
10 pages and the cover. More than one processor 80₁ may be required to complete such a job
11 request, or to complete the job request in the most cost-efficient and/or timely manner. The work
12 flow controller 70 (not shown in Figure 7) can determine which of the processors 80₁-80_N should
13 complete a specific process, and, if necessary, the order in which such processes should be
14 completed. The work flow controller 70 may poll the various processors 80₁ to determine which
15 may be used to complete the job request. The work flow controller 70 may then notify selected
16 processors 80₁ that a job request has been registered with the job ticket service 60.

17 For each job ticket 61₁ received, the job ticket service 60 creates a reference 72₁ to the
18 job ticket 61₁. The processor 80₁ may request access to the job ticket 61 in order to complete one
19 or more processes. In response, the job ticket service 60 provides the processor 80₁ with the job
20 ticket reference 72₁. The job ticket reference 72₁ is then used as an index to the job ticket 61₁.
21 The job ticket reference 72₁ may also be provided to other processors, such as the processor 80₂,
22 and to other clients, such as the client 31₂. The processor 80₂ and the client 31₂ may then access
23 the job ticket 61₁ at the same time as the processor 80₁ accesses the job ticket 61₁. This
24 simultaneous access allows different processes to be completed in parallel. In the example
25 illustrated in Figure 2, the processor 80₁ may complete some or all the processes for the inside
26 pages, and the processor 80₂ may complete the processes for the cover.

27 Figure 8 is a block diagram illustrating an example application of the control features of the
28 job ticket service 60. The job ticket 61₁ is referenced to the job content 51₁ by the job ticket ID
29 63, and information related to the job ticket 61₁ and the job content 51₁ is passed over the service
30 bus 41. The processors 80₁ can access the job content 51₁ and the job ticket 61₁ using the service

bus 41. In the illustrated example, the job ticket 61₁ refers to a job request 32 to print a brochure using the processes outlined in Figure 2. The processor 80₁ is designated by the work flow processor 70 to produce the inside pages of the brochure and the processor 80₂ is designated to produce the brochure cover. The processor 80₁ passes a job ticket access request to the job ticket service 60. The access request may include security information that allows the processor 80₁ to access the job ticket 61₁ and the corresponding content 51₁ or job. In response, the job ticket service 60 provides a job ticket reference 62₁ that is used by the processor 80₁ to access the job ticket 61₁. The processor 80₁ may use information in the job ticket 61₁ to access the content 51₁ stored in the job store 50. Since the processor 80₁ will produce only the inside pages, the processor 80₁ will not need access to all the information contained in the job ticket 61₁. Furthermore, because the job ticket 61₁ remains in the job ticket service 60, other entities, such as the processor 80₂, may continue to access the job ticket.

As the processor 80₁ completes various processes, the processor 80₁ may update the content 51₁ and the job ticket 61₁. Thus, the job ticket 61₁ may reflect the latest status of the job request 32. The status reports may indicate when a node in the node tree 10 is completed, when an interim deadline is completed, when another processor may be used to complete a process, and when all processing is complete. The status report may be included in a digital file that is used by the work flow controller 70, for example. The status report may also be included in a human-readable format, such as a pop-up window on a computer display screen. The processor 80₁ may receive the job ticket reference 72₁, and may complete all scheduled processes, returning the job ticket reference 72₁ to the job ticket service 60. The processor 80₁ may also send a copy of the job ticket reference 72₁ to the processor 80₂, so that the processor 80₂ may access the job ticket 61₁, and the content 51₁ and produce the brochure cover.

Figure 9 is a flowchart illustrating an operation 100 of the job ticket service 60. The operation 100 is based on completing the inside pages nodes shown in Figure 2. The operation 100 may be at least partly under the control of the work flow controller 70, or some equivalent device. The operation 100 assumes that a job request 32 (job ticket data and content) have been passed to the service center 40, and that a job ticket service 60 has been created. The operation 100 begins at start block 101. In review and assign processors block 105, the work flow controller 70 determines which processors 80₁ are able and available to complete the job. The

Figure 10 is a flowchart illustrating the routine 105 for developing a work flow and assigning processors to the work flow. The process starts in block 200. In block 205, the service center 40 receives a job request 32. The job request 32 may specify performance requirements, resources, and other parameters, and may include content 51, or a link to the content 51. In block 210, the work flow controller 70 defines a work flow to accomplish the tasks specified in the job request 32. The work flow may be represented by a node tree, such as the node tree 10 shown in Figure 2.

In block 230, the work flow controller 70 generates a job ticket 61 using the information provided by the job request 32, the work flow generated in block 210, and an appropriate job ticket template. The job ticket 61 is then stored in the job ticket service 60. Any content 51 may be stored in the job store 50.

The work flow controller 70 or the job ticket service 60 may create a job ticket notice, or other object, and may post the notice, block 250, at the service center 40 so that outside entities (e.g., the processors 80) may acquire sufficient information to bid on completion of the job ticket 61, or a branch 66 of the job ticket 61. In an alternative embodiment, the job ticket 61 may be posted at the service center 40. If the job ticket 61 is posted, the job ticket 61 may include mechanism to limit access to the job ticket or to limit access to certain portions of the job ticket 61. For example, the client extension 64 may not be accessible to the processors 80.

In block 270, the service center 40 receives bids from specific processors 80 and in block 290, the service center 40 evaluates the bids. In block 295, the service center 40 determines if the client 31 submitting the job request 32 intends to select the winning bid(s), or if the service center 40 makes the selection. If the client is to make the selections, in block 300, the service center 40 provides the bid information to the client 31. Then, in block 305, the service center 40 receives the selections from the client 31. If the service center 40 is to make the selections, in block 310, the service center 40 selects the winning bid(s). In block 315, the service center notifies the winning processors. The service center may also store the bid information with the corresponding job ticket 61. In block 320, the routine 105 ends.

Figure 11 is a flowchart illustrating the sub-routine 210 for defining a work flow. The sub-routine 210 starts in block 350. In block 355, the work flow controller 70 determines if the work flow will contain multiple branches. If the work flow will contain multiple branches, the work flow

controller 70 defines the branches, block 360. In block 365, the work flow controller 70 selects a branch for which resources and processes are to be defined. In block 370, the work flow controller 70 defines input resources for a first process, or node. In block 375, the work flow controller 70 defines the tasks to be completed for the first process. In block 380, the work flow controller 70 determines the output resources of the first process. In block 385, the work flow controller 70 determines if another process is required for the work flow or branch. In no additional processes are required, the work flow controller 70 determines if another branch is to be defined, block 390. If another branch is to be defined, the work flow controller 70 selects another branch, block 365, and the sub-routine 210 continues. If another branch is not to be defined, the sub-routine ends 210, block 395. The results of the work flow definition may be incorporated into the job ticket 61 (see Figure 10, block 230).

Figure 12 is a flow chart illustrating the sub-routine 250 of posting a job ticket notice or job ticket. The sub-routine 250 starts in block 400. In block 405, the work flow controller 70 determines if the work flow associated with the job ticket 61 includes multiple branches. If the work flow does not include multiple branches, the work flow controller posts the job ticket notice listing the single branch, block 410. If the work flow includes multiple branches, the work flow controller 70 posts the job ticket notice with multiple branches, block 420. The sub-routine 250 then ends.

Figure 13 is a flow chart illustrating the sub-routine 290 for evaluating bids. The sub-routine starts in block 440. In block 445, the bidding service 90 selects a first bid for analysis. In block 450, the bidding service 90 determines if the client 31 has supplied any evaluation criteria or requirements. If the client has not supplied evaluation requirements, the bidding service 90 compares the selected bid to a set of standard, minimum performance requirements, which may be industry-standard requirements block 455. In block 460, the bidding service 90 determines if the bid meets the minimum performance requirements. If the bid does not meet the minimum performance requirements, the bid is rejected, block 475. If the bid is rejected, the bidding service 90 determines if additional bids were submitted, block 495. If additional bids were submitted, the bidding processor 90 returns to block 445 and selects the next bid for evaluation.

In block 450, if the client 31 has supplied performance requirements, the bidding service 90 compares the selected bid to the client-supplied performance requirements, block 465. In

block 470, the bidding service 90 determines if the selected bid meets the minimum criteria of the client-supplied performance requirements. If the minimum criteria are not met, the bidding service 90 rejects the bid, block 475.

In blocks 470 and 460, if the minimum criteria are met, the bidding service 90 determines if the client 31 has supplied an evaluation algorithm. If the client 31 has not supplied an evaluation algorithm, the bidding service applies a standard evaluation algorithm, which may be an industry-standard algorithm, block 485. If the client has supplied an evaluation algorithm, the bidding service 90 applies the client-supplied evaluation algorithm, block 490. The bidding service 90 may then store the results of the algorithm pending evaluation of all bids.

In block 495, the bidding service 90 determines if any bids remain to be evaluated. If additional bids remain, the sub-routine 290 returns to block 445, and the bidding service selects the next bid for evaluation. In block 495, if no additional bids remain for evaluation, the bidding service 90 ranks the bids, block 500. The sub-routine 290 then ends, block 505.

Figure 14 is a flowchart illustrating the routine 130 for providing access to a job ticket 61. The routine 130 begins in block 510. In block 515, the job ticket service 60 receives a job ticket reference 72 from a processor 80, and retrieves the corresponding job ticket 61, block 520.

In block 525, the job ticket service 60 compares the processor identification to processors listed in the job ticket 61 or branches 66 of the job ticket 61. The job ticket service 60 determines if the selected branches 66 are locked, block 530. If the selected branches 66 are not locked, the job ticket service 60 copies the selected branches 66 to the processor 80, block 535. In block 550, the job ticket service 60 then determines if the selected branches 66 require locking. If the selected branches do not require locking, the routine 130 ends, block 560. If the selected branches 66 require locking, the job ticket service 60 locks the selected branches 66, block 555. The routine 130 then ends, block 560.

In block 530, if the selected branches 66 are locked, the job ticket service 60 determines if the processor 80 intends to modify information in the selected branches 66, block 540. If the processor 80 will not modify the selected branches 66, the job ticket service 60 may provide an error message, block 545. If the selected branches 66 will be modified, the job ticket service 60 may unlock the selected branches 66.

Figure 15 is a flow diagram of a method for allowing access to a job ticket 61. The method may execute as part of the routine 115 shown in Figure 9. The method starts with block 600. In block 605, the authentication server 94 receives authentication information from a processor 80 and retrieves a job ticket 61 corresponding to a job ticket reference 72 possessed by the processor 80. At this stage of the process, the job ticket 61 (excluding the public key signature field 67) contains two information fields, the framework 62 and the client extension 64. The framework 62 contains information such as the service ID, client IP address, expiration date and time, and processor authorization, as previously described. The client extension 64 contains information such as credit card number and zip code, also previously described. The information in the job ticket 61 (excluding the public key signature field 67) is then, for example, optionally hashed using, for example, MD5 protocol, and encrypted with a public key encryption system, block 610, generating a hash number, block 615. Other hashing or encryption techniques may also be used. The hash number is representative of the specific information contained in the job ticket 61. The hash number generated in block 615 is then encrypted using a standard public key encryption system, block 620. Encrypting the hash number with a private key prevents any user without knowledge of the public key from modifying the job information. In block 625, the job ticket 61 and the encrypted hash number are concatenated to generate the completed job ticket 61. Hence, the completed job ticket 61 information fields: 1) the framework 62, 2) the client extension 64, and 3) the public key signature (encrypted hash number) 67. The method then ends, block 630.

In the illustrated embodiments, the service center 40, and its sub-components, including the work flow controller 70 and the job ticket service 60, for example, may be implemented as a single, special purpose integrated circuit (e.g., an ASIC) having a main or central processor section for overall, system-level control, and separate circuits dedicated to performing various different computations, functions and other processes under control of the central processor section. Those skilled in the art will appreciate that the service center 40 may also be implemented using a plurality of separate, dedicated or programmable integrated or other electrical circuits or devices (e.g., hardwired electronic or logic circuits such as discrete element circuits, or programmable logic devices such as PLDs, PLAs, or PALs). The service center 40 may also be implemented using a suitably programmed general purpose computer, e.g., a microprocessor, microcontroller or other processor device (CPU or MPU), either alone or in conjunction with one or more peripheral (e.g.,

1 integrated circuit) data and signal processing devices. In general, any device or assembly of
2 devices on which a finite state machine capable of implementing the flowcharts shown in Figures
3 9-15 can be used as the service center 40, or its sub-components.

4 The terms and descriptions used herein are set forth by way of illustration only and are not
5 meant as limitations. Those skilled in the art will recognize that many variations are possible within
6 the spirit and scope of the invention as defined in the following claims, and their equivalents, in
7 which all terms are to be understood in their broadest possible sense unless otherwise indicated.

09073196 060501
T05090 " 96TE/860